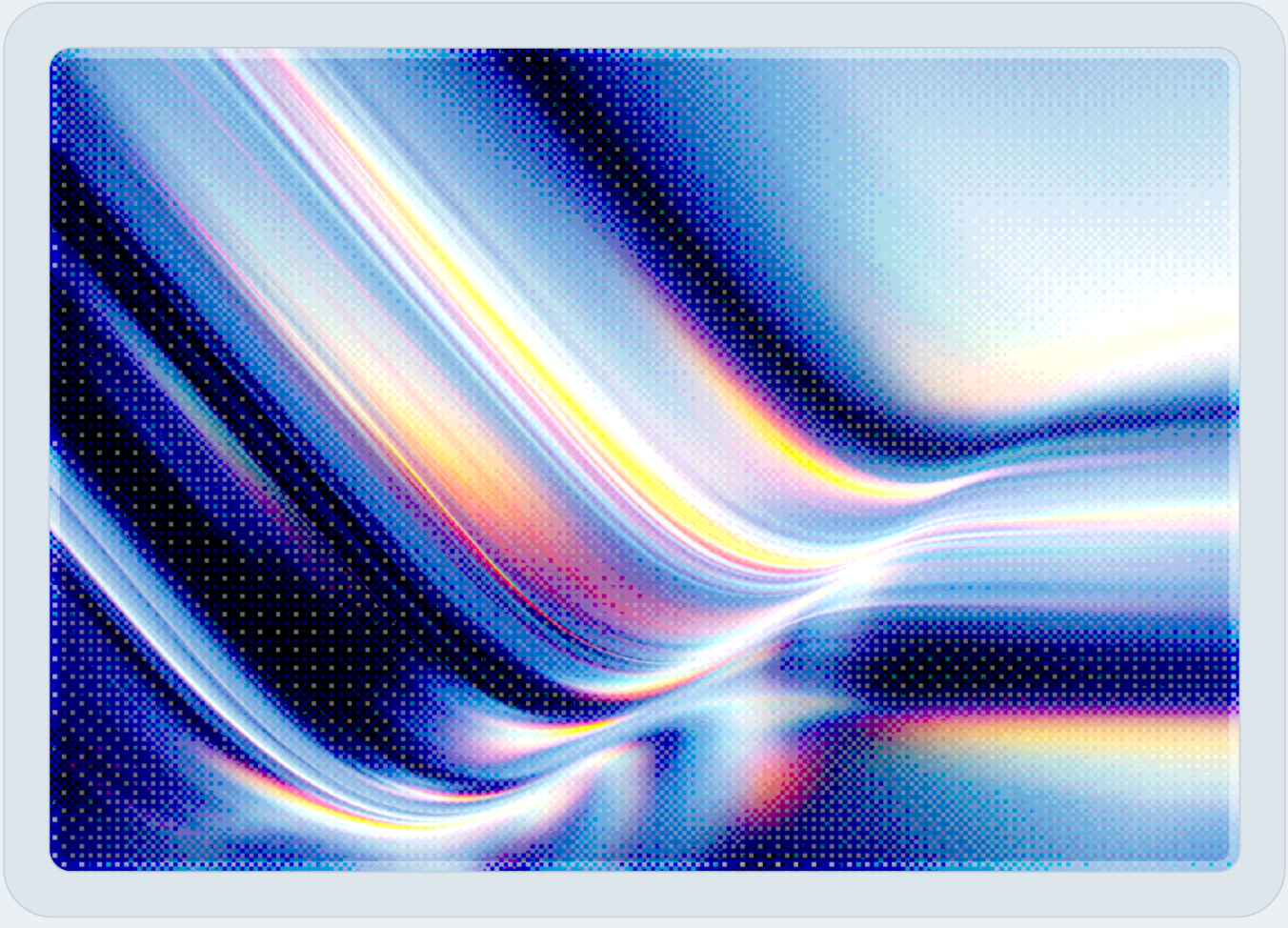
CSAI Foundation | Cloud Security Alliance

Sub-Hour Exploitation of AI Inference Infrastructure

The Collapsing Window Between Disclosure and Active Attack

2026-05-17

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- The window between public vulnerability disclosure and active exploitation of AI inference and agent infrastructure has compressed to hours — in the case of CVE-2026-44338 (PraisonAI), to under four hours.
- Inference servers including Ollama (CVE-2026-7482, "Bleeding Llama"), vLLM (CVE-2025-30165), and model orchestration platforms including Flowise (CVE-2025-59528) carry critical-severity flaws that attackers have actively weaponized in 2025 and 2026.
- Mandiant's M-Trends 2026 data shows the mean time to exploit has gone negative — exploitation is now occurring, on average, seven days before a vendor patch is available, leaving patch-dependent defenses structurally inadequate [1].
- AI inference services are frequently deployed without authentication, network segmentation, or audit logging — a pattern documented across Ollama, PraisonAI, and MCP server exposures — reducing attacker effort that historically required days of manual reconnaissance to minutes of automated scanning.
- Organizations must treat AI inference infrastructure with the same urgency and control maturity applied to production databases and identity systems.

Background

The rapid deployment of on-premises and cloud-hosted AI inference infrastructure has created a new and underprotected attack surface. Platforms such as Ollama, vLLM, Flowise, LangChain, and Semantic Kernel are now embedded in enterprise automation pipelines, CI/CD workflows, and customer-facing applications — often deployed by engineering teams without security review and with default configurations that prioritize developer convenience over access control.

This deployment pattern reproduces a failure mode well-documented in earlier infrastructure security research [17]. When Redis, Elasticsearch, and Docker daemon sockets were first widely adopted, operators frequently exposed them without authentication, and attackers rapidly identified and exploited these gaps. AI inference infrastructure is following the same trajectory, with compounding risk: unlike a

misconfigured database, a compromised inference endpoint can exfiltrate ongoing conversation context, system prompts containing business logic, and API credentials stored in environment variables, while also providing a foothold into the broader compute environment.

Historically, defenders benefited from a meaningful gap — often measured in weeks or months — between public vulnerability disclosure and widespread exploitation. For AI inference infrastructure specifically, the evidence suggests this window has compressed dramatically. CSA's 2026 analysis of the AI vulnerability landscape, building on "The AI Vulnerability Storm" expedited briefing [2], identifies accelerated weaponization as among the most consequential near-term threats to AI infrastructure. Mandiant's M-Trends 2026, drawn from over 500,000 hours of frontline incident investigations, found that exploitation began an average of seven days before the corresponding patch was available [1] — meaning organizations that rely on patch-then-protect strategies are consistently behind.

Security Analysis

The Four-Hour Proof of Concept: CVE-2026-44338 (PraisonAI)

On May 11, 2026, researchers published an advisory for CVE-2026-44338, an authentication bypass vulnerability in PraisonAI's legacy Flask API server. The flaw stems from a configuration default of `AUTH_ENABLED = False` and `AUTH_TOKEN = None`, leaving the agent API's `/agents` enumeration endpoint and `/chat` execution endpoint fully unauthenticated [3][4]. Sysdig researchers monitoring network traffic observed the first automated scan — a request from a client identifying itself as `CVE-Detector/1.0` — precisely three hours and 44 minutes after the advisory became public at 13:56 UTC [4].

The practical consequence is not theoretical: a simple, unauthenticated POST to `/chat` forces the victim's infrastructure to execute the system's local `agents.yaml` workflow, draining costly external API quotas and exfiltrating any output data returned. The CVSS base score of 7.3 does not capture the deployment-context impact when the affected service is connected to sensitive data pipelines or external APIs with metered billing; in those environments, operational consequences are substantially higher. PraisonAI has patched this in version 4.6.34, but the incident illustrates a broader pattern: AI agent frameworks developed under rapid iteration cycles routinely ship with insecure-by-default configurations, and adversaries have automated the process of scanning for newly disclosed CVEs.

Inference Server Memory Disclosure: CVE-2026-7482 ("Bleeding Llama")

Cyera researchers disclosed CVE-2026-7482 in May 2026, a CVSS 9.1 heap out-of-bounds read in Ollama's GGUF model loader [5][6]. The root cause is a missing bounds check in Ollama's tensor parsing code: the application reads a declared tensor offset and size without verifying they correspond to the file's actual length. By crafting a model file with a misaligned float-16 source tensor targeting a float-32 destination, an attacker can trigger a lossless conversion path that copies heap memory beyond the intended bounds into the response stream.

The attack requires only three unauthenticated API calls and generates no error entries in the application log [5]. Data recoverable from leaked heap memory includes active user and system prompts from concurrent sessions, environment variables — potentially including API keys, database credentials, and proprietary system instructions — and residual content from previously loaded models. As of early May 2026, approximately 300,000 Ollama server instances were exposed on the public internet [6], and many more exist on corporate networks where implicit trust assumptions may create equivalent exposure. The vulnerability was patched in Ollama v0.17.1, released on February 25, 2026, but Ollama did not flag the release as a security fix, and disclosure timelines were significantly delayed — a pattern Cyera's researchers attributed in part to CVE assignment processing delays [5].

The Bleeding Llama case illustrates a transparency risk that can exist in the AI infrastructure ecosystem: security-relevant fixes may be shipped within general-purpose release notes without the indicators that security teams use to triage urgency. Organizations relying on standard vulnerability feeds experienced a months-long blind spot between patch availability and actionable awareness.

Supply Chain Propagation: The ShadowMQ Pattern

Researchers at Oligo Security disclosed over 30 critical vulnerabilities in November 2025 across a cluster of major AI inference engines — including Meta's Llama framework (CVE-2024-50050), vLLM (CVE-2025-30165), NVIDIA TensorRT-LLM (CVE-2025-23254), Modular Max Server (CVE-2025-60455), and Microsoft Sarathi-Serve [7][8]. The vulnerabilities share a common root cause, now designated the "ShadowMQ" pattern: unsafe use of ZeroMQ's `recv_pyobj()` method, which deserializes incoming messages using Python's `pickle` module without validation. When the corresponding network interface is reachable by an attacker, a single maliciously crafted pickle object can achieve full remote code execution on the host.

The propagation mechanism is as instructive as the vulnerability itself. Code headers in vLLM's affected module documented "Adapted from Llama Stack," indicating that the insecure pattern propagated from the Llama Stack codebase — suggesting that no independent security review was applied to the ported code before it shipped. This represents a structural risk inherent to the rapid-iteration culture of AI

tooling: frameworks are assembled from shared components under competitive time pressure, and a vulnerability introduced at one layer propagates across the ecosystem before any component undergoes formal security audit. CVE-2025-30165 affects vLLM versions 0.5.2 through 0.8.5.post1 under the V0 engine, with the V1 engine default in version 0.10.0 providing partial — but not complete — remediation, according to Oligo Security's advisory [7].

AI Agent Builders: Critical RCE in Production Tooling

Flowise, an open-source low-code platform used to build AI agent workflows, has experienced three separate in-the-wild exploitations since 2025. The most severe, CVE-2025-59528 (CVSS 10.0), is a JavaScript code injection flaw in Flowise's CustomMCP node [9][10]. The node allows users to configure connections to external MCP servers but executes arbitrary JavaScript with no validation, effectively exposing a fully authenticated remote code execution primitive to anyone who can reach the Flowise API. VulnCheck's Canary network detected the first exploitation attempts in April 2026, originating from a single Starlink IP address [9]. As of that disclosure, over 12,000 Flowise instances remained exposed on the public internet [9] — a population that, given Flowise's positioning as a production agent builder, likely includes enterprise deployments. The disclosure noted that CVE-2025-59528 is the third Flowise CVE to see in-the-wild exploitation, a pattern indicating that Flowise's attack surface is actively profiled by threat actors.

Microsoft's disclosure of CVE-2026-25592 and CVE-2026-26030 in the Semantic Kernel .NET and Python SDKs, respectively, demonstrates that the same class of vulnerability affects frameworks from major enterprise vendors [11][12]. CVE-2026-25592 (CVSS 10.0) exploits a path traversal flaw in `DownloadFileAsync`, an internal helper method that was accidentally decorated as `[KernelFunction]`, exposing it to the language model without authorization checks. An attacker who can influence a single agent prompt — through a poisoned document, email summary, or user message — can direct the agent to write arbitrary files outside the Azure Container Apps sandbox. Combined with access to a startup directory or CI hook, this yields full remote code execution on the agent's host with no authentication and no network position requirement beyond reaching the agent itself [11].

The Unauthenticated Exposure Baseline

The vulnerabilities above are individually severe, but they operate against a substrate of baseline exposure that amplifies their impact. Trend Micro research, aggregated in a 2026 analysis of AI agent security risks, identified more than 8,000 MCP servers accessible on the public internet, 492 of which had neither client authentication nor traffic encryption [13]. SecurityScorecard data from the same report found OpenClaw, an open-source AI desktop agent platform, showing 135,000 instances accessible with

insecure defaults [13]. This exposure profile is not the result of operator negligence alone; many inference and agent platforms do not ship with authentication enabled by default, and deployment guidance frequently does not foreground security configuration.

When authentication is absent, disclosed vulnerabilities become accessible to the broadest possible adversary population, dramatically reducing the attacker effort required for exploitation. The PraisonAI case illustrates how rapidly specialized tooling can target newly disclosed CVEs — the appearance of a scanner self-identifying as `CVE-Detector/1.0` within hours of disclosure suggests that CVE-specific tooling is being actively deployed against AI infrastructure at a pace that leaves little margin for delayed response.

Recommendations

Immediate Actions

The most consequential near-term step for any organization operating AI inference or agent infrastructure is to conduct an inventory of exposed services and apply network-level access controls immediately, independent of patching status. Ollama, vLLM, and Flowise should never be reachable from the public internet without an authenticated reverse proxy or API gateway. Internal exposure should be bounded to the specific services and identities that require access, enforced at the network layer rather than relying on application-level authentication alone.

Organizations should prioritize the following patch actions as of this writing. For Ollama: upgrade to version 0.17.1 or later to remediate CVE-2026-7482. For Semantic Kernel: upgrade to .NET SDK version 1.71.0 and Python SDK version 1.39.4 to remediate CVE-2026-25592 and CVE-2026-26030. For Flowise: upgrade to version 3.0.6 or later to remediate CVE-2025-59528. For PraisonAI: upgrade to version 4.6.34 or later to remediate CVE-2026-44338. For vLLM users running the V0 engine: evaluate migration to the V1 engine and monitor for updated guidance on CVE-2025-30165 remediation completeness, as Oligo Security's advisory notes that the V1 engine provides partial but not complete remediation [7].

Given the evidence that exploitation windows are collapsing to hours, subscribe to vulnerability notification feeds specific to AI tooling — including GitHub Security Advisories for the frameworks in use — and treat any critical-severity AI infrastructure CVE as requiring same-day assessment and remediation planning, not queue entry into a monthly patch cycle.

Short-Term Mitigations

Authentication defaults across the AI tool stack require active review. Every inference API, agent framework, and MCP server should have authentication enabled and configured before it is reachable by any network-connected consumer, including internal services. API keys and tokens used by inference infrastructure should be stored in a secrets manager, not in environment variables, given that heap-memory disclosure vulnerabilities like Bleeding Llama make environment variable theft a realistic attack outcome against unpatched services.

Audit logging and anomaly detection should be applied to AI inference endpoints at a maturity comparable to that applied to production databases. Unusual patterns — elevated token consumption, unexpected API calls to external endpoints, requests from unfamiliar network segments — are often the first indicators of credential theft or unauthorized agent execution. The PraisonAI exploitation pattern, in which attackers force the system to execute local agent workflows and exhaust external API quotas, would be detectable through billing anomaly monitoring if that capability were in place.

Organizations using AI agent builders for production workflows should apply the principle of least privilege to the functions and tools exposed to the model. The Semantic Kernel CVE-2026-25592 exploited an accidentally-exposed helper function. A regular audit of which methods are decorated as model-callable — and whether each requires the access it is granted — would reduce this class of risk across any framework that exposes tools to a language model.

Strategic Considerations

The ShadowMQ disclosure illustrates that copy-paste vulnerability propagation is a structural feature, not an anomaly, of the current AI tooling ecosystem. Organizations should require software composition analysis (SCA) for AI inference dependencies and treat AI frameworks with the same supply chain scrutiny applied to other critical infrastructure components. The presence of an explicit attribution comment ("Adapted from Llama Stack") in a vulnerable module is a useful reminder that provenance information is already in the code — security teams should be reading it.

The AI infrastructure attack surface is expanding faster than security controls are being applied to it. Enterprise security teams that are not actively inventorying AI tooling deployments — including shadow deployments initiated by engineering and data science teams outside security review — are operating with an incomplete picture of their exposure. A governance framework that gates AI infrastructure deployments on a baseline security checklist, including authentication, network segmentation, and vulnerability notification enrollment, is a proportionate response to the current threat landscape.

Finally, the negative mean time to exploit documented by Mandiant [1] represents a structural shift that invalidates patch-then-protect as a primary risk mitigation strategy. Defense-in-depth, including network controls that limit the reachability of vulnerable services and monitoring that detects anomalous use before exploitation is confirmed, must become the baseline posture for AI infrastructure — not an enhancement applied after incidents occur.

CSA Resource Alignment

This research note aligns with several active CSA frameworks and publications. The MAESTRO (Multi-Agent Environment, Security, Threat, Risk, and Outcome) framework [14] provides a seven-layer reference architecture for agentic AI threat modeling, covering Foundation Models, Data Operations, Agent Frameworks, Deployment Infrastructure, and the broader Agent Ecosystem. The vulnerabilities documented here affect MAESTRO's Layer 3 (Agent Frameworks), Layer 4 (Deployment and Infrastructure), and Layer 7 (Agent Ecosystem), underscoring the need to apply threat modeling at each layer independently rather than treating AI infrastructure as a monolithic stack.

The AI Controls Matrix (AICM) v1.0.3 [15], particularly its implementation guidelines for Application Providers and Orchestrated Service Providers, addresses the shared responsibility boundaries relevant to inference infrastructure deployments. Controls in the AI Supply Chain Security and Deployment and Infrastructure domains are directly applicable to the ShadowMQ and Flowise vulnerability classes. CSA's "Securing Autonomous AI Agents" survey [16], drawing on data collected through late 2025, found that AI agent security incidents were already common in enterprises — the vulnerability cases documented here provide concrete technical grounding for those survey findings.

"The AI Vulnerability Storm" expedited briefing [2], published by CSA in April 2026, addresses the broader acceleration of AI-driven vulnerability discovery and exploitation and provides a Mythos-ready security program framework applicable to organizations responding to the threat landscape described in this note.

References

[1] Mandiant / Google Cloud. "M-Trends 2026: Data, Insights, and Strategies From the Frontlines." Google Cloud Blog, March 2026.

[2] CSA AI Safety Initiative. "The AI Vulnerability Storm: Building a Mythos-Ready Security Program." Cloud Security Alliance, April 2026.

[3] The Hacker News. "PraisonAI CVE-2026-44338 Auth Bypass Targeted Within Hours of Disclosure." The Hacker News, May 2026.

[4] Sysdig. "CVE-2026-44338: PraisonAI Authentication Bypass in Under 4 Hours and the Growing Trend of Rapid Exploitation." Sysdig Blog, May 2026.

[5] Cyera Research. "Bleeding Llama: Critical Unauthenticated Memory Leak in Ollama." Cyera, May 2026.

[6] CybersecurityNews. "Critical Ollama Memory Leak Vulnerability Exposes 300,000 Servers Globally." Cybersecurity News, May 2026.

[7] Oligo Security. "ShadowMQ: How Code Reuse Spread Critical Vulnerabilities Across the AI Ecosystem." Oligo Security Blog, November 2025.

[8] Rescana. "ShadowMQ Vulnerabilities: Over 30 Critical Flaws in Meta Llama, NVIDIA TensorRT-LLM, vLLM, and Other AI Inference Engines." Rescana, November 2025.

[9] The Hacker News. "Flowise AI Agent Builder Under Active CVSS 10.0 RCE Exploitation; 12,000+ Instances Exposed." The Hacker News, April 2026.

[10] SonicWall. "FlowiseAI Flowise RCE via CustomMCP Node CVE-2025-59528." SonicWall Blog, 2026.

[11] Microsoft Security Blog. "When Prompts Become Shells: RCE Vulnerabilities in AI Agent Frameworks." Microsoft, May 2026.

[12] Particula. "Semantic Kernel CVE-2026-25592: How Prompt Injection Became RCE." Particula Tech Blog, 2026.

[13] Cyberdesserts. "AI Agent Security Risks 2026: MCP, OpenClaw and Supply Chain." Cyberdesserts Blog, 2026.

[14] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA Blog, February 2025.

[15] Cloud Security Alliance. "AI Controls Matrix (AICM)." CSA, 2025.

[16] Cloud Security Alliance. "Securing Autonomous AI Agents: Survey Report." CSA, 2026.

[17] SecurityWeek. "8,000 Unprotected Redis Instances Accessible From Internet." SecurityWeek, 2015.